**EXERCÍCIOS DE POO**

* **Utilizar os seguintes repositórios para consulta**
  + **POO PARTE 1 -** [**https://github.com/rafaelq80/cookbook\_java/blob/main/10.md**](https://github.com/rafaelq80/cookbook_java/blob/main/10.md)
  + **POO PARTE 2 -** [**https://github.com/rafaelq80/cookbook\_java/blob/main/11.md**](https://github.com/rafaelq80/cookbook_java/blob/main/11.md)
  + **ÍNDICE GERAL -** [**https://github.com/rafaelq80/cookbook\_java/blob/main/README.md**](https://github.com/rafaelq80/cookbook_java/blob/main/README.md)

1. O que é polimorfismo e como ele é implementado em Programação Orientada a Objetos?

Resposta: polimorfismo é a habilidade de objetos de classes diferentes responderem a mesma mensagem de diferentes maneiras, ou seja, várias formas de responder à mesma mensagem. O Polimorfismo é a capacidade de um objeto decidir que método aplicar a si mesmo, embora a mensagem possa ser a mesma, os objetos podem responder diferentemente.

* Resposta: Ao enviar uma mensagem que pede para uma **subclasse** aplicar um **método** usando certos parâmetros, a subclasse verifica se ela tem ou não um método com esse nome e exatamente com os mesmos parâmetros.
* Se tiver, usa-o.
* Caso contrário: a **superclasse** torna-se responsável pelo processamento da mensagem e procura por um método com esse nome e esses parâmetros.
* Se encontrar, chama esse método.

1. Qual é a diferença entre encapsulamento e abstração em Programação Orientada a Objetos?

Resposta: **Encapsular**, nada mais é do que proteger membros de uma classe de acesso externo, permitindo somente sua manipulação de forma indireta.

Resposta: **Abstrato:** que não é concreto; que resulta da abstração, que opera unicamente com ideias, com associações de ideias, não diretamente com a realidade sensível, que possui alto grau de generalização.

1. O que é herança múltipla e por que muitas linguagens de programação não a suportam?

* Resposta: **Herança Múltipla:** É a capacidade de uma classe possuir mais de uma superclasse e herdar os atributos e métodos de todas as superclasses. Java não implementa Herança Múltipla nativamente, como a Linguagem C++, por exemplo. Para simular a herança múltipla em Java, usa-se **Interfaces**.
* Resposta: Java não implementa Herança Múltipla nativamente

1. O que é o padrão de design Singleton e por que ele é frequentemente usado em Programação Orientada a Objetos?

Resposta:permite criar objetos únicos para os quais há apenas uma instañcia.

1. O que são classes abstratas e para que são usadas?
2. O que é sobrecarga de operadores em Programação Orientada a Objetos e como ela é implementada?
3. O que são coleções em Programação Orientada a Objetos e como elas diferem de arrays?
4. O que é injeção de dependência e qual é sua importância em Programação Orientada a Objetos?
5. O que é acoplamento e coesão em Programação Orientada a Objetos e qual é a diferença entre eles?
6. O que é uma interface em Programação Orientada a Objetos e como ela é usada?
7. O que é o padrão de design Factory Method e como ele é usado em Programação Orientada a Objetos?
8. Qual é a diferença entre uma classe e um objeto em Programação Orientada a Objetos?
9. O que são construtores em Programação Orientada a Objetos e para que são usados?
10. O que são eventos em Programação Orientada a Objetos e como eles são usados?
11. O que é reflexão em Programação Orientada a Objetos e como ela é usada?